PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : H03H 17/06, 17/02, G06F 15/332

A1

(11) International Publication Number: WO 94/01933

(43) International Publication Date: 20 January 1994 (20.01.94)

(21) International Application Number: PCT/AU93/00330

(22) International Filing Date: 5 July 1993 (05.07.93)

(30) Priority data:

PL 3386	7 July 1992 (07.07.92)	AU
PL 3387	7 July 1992 (07.07.92)	AU
PL 8357	20 April 1993 (20.04.93)	AU

(71) Applicant *(for all designated States except US)*: LAKE DSP PTY. LIMITED [AU/AU]; 1/166 Maroubra Road, Maroubra, NSW 2035 (AU).

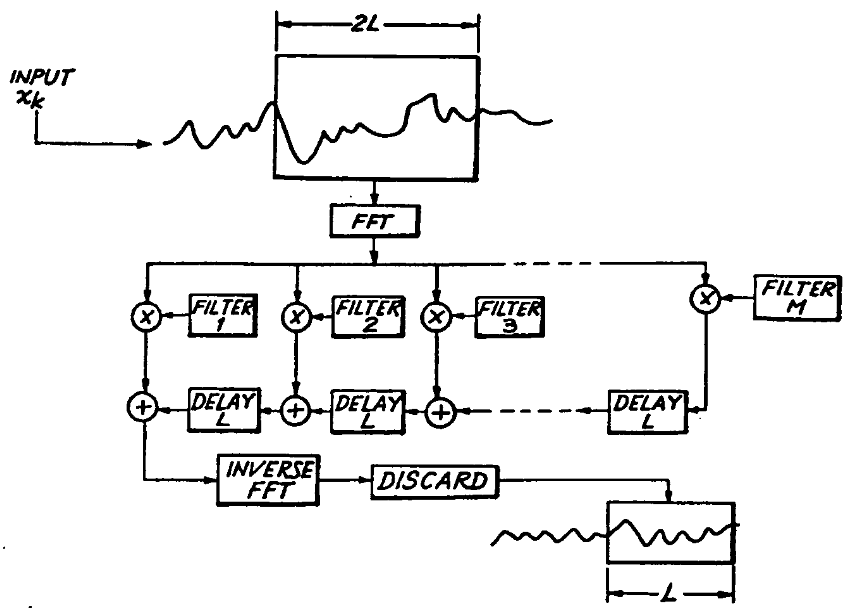
(72) Inventor; and
(75) Inventor/Applicant *(for US only)* : McGRATH, David, Stanley [AU/AU]; 16 Caerleon Crescent, Randwick, NSW 2031 (AU).

(74) Agent: MAXWELL, Peter, Francis; Peter Maxwell & Associates, Blaxland House, 5 Ross Street, North Parramatta, NSW 2151 (AU).

(81) Designated States: AT, AU, BB, BG, BR, BY, CA, CH, CZ, DE, DK, ES, FI, GB, HU, JP, KP, KR, KZ, LK, LU, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SK, UA, US, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: DIGITAL FILTER HAVING HIGH ACCURACY AND EFFICIENCY



(57) Abstract

Apparatus for and methods of operation of digital filters and certain other electronic digital signal processing devices are provided to improve the accuracy and efficiency of filtering. Particularly, the apparatus and method includes a digital filter with a long impulse response and lower latency, built by operating a numbr of small component filters (F1, F2, F3) in parallel and combining their outputs by addition, with each component filter operating with a different delay such that the net operation of the ensemble of said component filters is the same as a single filter with a longer impulse response, and the latency of the ensemble is equal to the shortest latency of the component filters. At least one group of the component filters is implemented using a Discrete Fourier Transform technique. A Fourier transform processor adapted to efficiently transform strings of real data is also described.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NE	Niger
BE	Belgium	GN	Guinea	NL	Netherlands
BF	Burkina Faso	GR	Greece	NO	Norway
BG	Bulgaria	HU	Hungary	NZ	New Zealand
BJ	Benin	IE	Ireland	PL	Poland
BR	Brazil	IT	Italy	PT	Portugal
BY	Belarus	JP	Japan	RO	Romania
CA	Canada	KP	Democratic People's Republic of Korea	RU	Russian Federation
CF	Central African Republic			SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovak Republic
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	UA	Ukraine
DE	Germany	MG	Madagascar	US	United States of America
DK	Denmark	ML	Mali	UZ	Uzbekistan
ES	Spain	MN	Mongolia	VN	Viet Nam
FI	Finland				

DIGITAL FILTER HAVING HIGH ACCURACY AND EFFICIENCY

FIELD OF THE INVENTION

The present invention relates to the art of electronic signal processing, and more particularly but not exclusively, to an electronic filtering environment wherein relatively high accuracy and efficiency is desired and a relatively short flow-through delay (termed "latency") is desired.

DESCRIPTION OF THE PRIOR ART

With reference to Fig. 1, electronic filters are utilised to modify the characteristics of an incoming electronic signal so as to provide an output signal which is modified in some defined fashion. In the case of Fig. 1 a "notch" filter is illustrated wherein, in the frequency domain, frequencies in the spectrum of the incoming signal S1 are attenuated in the F_1 to F_2 band so as to produce output signal S2.

Such filters can be implemented from entirely analog components although, in more recent times, there is a preference, in many circumstances, to implement the filter in a digital fashion. Digital implementation can be by means of dedicated digital circuitry or by means of computers (micro processors) programmed to operate as a filter.

Filters have many applications in the field of electronic modelling of real world conditions. For example, filters can be used to provide a model of the acoustic characteristics of rooms or halls. Filters are also used to model deficiencies in systems so as to apply appropriate correction factors for the purpose of removing (cancelling)

imperfections in signals caused by the deficiencies.

Frequently it is desirable that such processing take place in "real time". Also, it is desirable that there is effectively no delay in filtering of a signal generated in a real/live environment so that the modelling/correcting steps performed by the filter are, to all intents and purposes, without any delay being perceptible to the end user.

To achieve this the delay introduced by the filter F while it performs its filtering function must be reduced to a negligible figure. That is, the time when signal S1 first presents to filter F and the time when the results of the filtering by filter F of the first incoming portion of signal S1 become available at the output of the filter S2 must be almost the same. The delay between these two events is hereinafter referred to as the "latency" of the filter system.

Where the filter F is implemented in a digital manner it may first be necessary to sample the incoming signal S1 (via an analog to digital converter) then perform the filtering function and then convert the digital signal back to an analog signal (by means of a digital to analog converter). The sampling process takes samples of the incoming signal at discrete time intervals t_i. The time between each sample is usually the same.

The sampling processing itself introduces finite delays into the system. Additionally, where the filter is implemented by one of the popular fast convolution techniques there is a delay introduced which, in very broad terms, is

proportional to the accuracy (or length) of the filter.

Mathematically, the filtering operation (that is, the step of imposing the filter characteristic upon the incoming signal S1 so as to produce outgoing signal S2) is known as "convolution" in the time domain. The step of convolution in the time domain becomes a multiply operation in the frequency domain. That is, if the incoming signal S1 is first sampled, then Fourier transformed into the frequency domain, the frequency response of the filter F is vector multiplied with the Fourier transform of the signal S1. The signal is then inverse Fourier transformed to produce a sampled (convolved) output (which can be converted back to analog form if required).

Figure 2 shows the way a convolver (also known as a Finite Impulse Response (FIR) filter) has its impulse response $\{a_k\}$ measured (for a convolver operating on a treatment of sampled data). For a physical filter, a_k is zero for all $k<0$. For a general input sequence $\{x_k\}$, the filter's output $\{Y_k\}$ is defined as:

$$Y_k=\sum_{i=0}^{\infty} a_i\, x_{k-i} \qquad (1)$$

A linear filter such as this has a measurable latency, d, defined as:-

$$\begin{aligned} &a_d \neq 0, \text{ and} \\ &a_k = 0 \text{ for all } k < d \end{aligned} \qquad (2)$$

In other words, a_d is the first non-zero value in the output sequence. The latency d is never negative in a physical system. In a similar fashion, for a Finite Impulse Response Filter, we can determine which is the LAST non-zero value in the output sequence. This will give us the length of the impulse response. If we call the length l, then this means that a_{d+l-1} is the last non-zero value in the output sequence (see Figure 3).

Typical schemes for implementing FIR filters fall into two categories:-

1. Time domain filters that compute each output sample as each new input sample arrives, thus allowing latencies as low as d=0 or d=1. Typical filter lengths (l) are short.
2. Fast convolution filters that compute a number of output samples in a block. Typical filter lengths (l) can be very long. The lowest achievable latency is usually related to the filter length, $d \approx l \div K$ or

$$K \approx l \div d \qquad (3)$$

where K is a measure of the efficiency of the particular algorithm used. A typical value of K, for the commonly used fast convolution algorithms such as illustrated in Figs. 4 and 5, is 0.5.

<u>BRIEF DESCRIPTION OF THE INVENTION</u>

It is an object of at least a preferred embodiment of the present invention to provide a method and apparatus for performing relatively long convolutions on digital sampled

data so as to provide relatively higher efficiency for a given length than is ordinarily produced with other methods.

In this specification it is assumed that the filter characteristics can be modelled as approximately linear so that the principles of superposition can be applied.

Accordingly, in one broad form of the invention there is provided a filter with a long impulse response and low latency, built by operating a number of smaller component filters in parallel and combining their outputs by addition, with each component filter operating with a different delay such that the net operation of the ensemble of said component filters is the same as a single filter with a longer impulse response, and the latency of the ensemble is equal to the shortest latency of the said component filters.

Preferably the component filters are implemented in different ways, with some filters adapted to provide low latency, and other filters adapted to provide longer filter lengths, such that the ensemble filter provides both low latency and long impulse response characteristics.

Preferably one or more of the component filters is implemented as a time-domain finite impulse response filter (built with multiply and add operations) and the remainder are implemented using a fast convolution method, such that the time-domain filter(s) provides the lowest latency portion of the ensemble impulse response, and the fast-convolution filter(s) provide the longer filter components.

Preferably the fast-convolution filters are built using the Discrete Fourier Transform or the Fast Fourier Transform.

In an alternative preferred form the fast-convolution filters are built using the Modified Discrete Fourier Transform as described in this specification.

Preferably a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one transform operation is performed for each group, so that the component filters in each group share the transformed output from the respective transform operation.

In an alternative preferred form a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one inverse transform operation is performed for each group, so that the output from the component filters in each group is summed before being passed to the respective inverse transform operation.

In a further alternative preferred form a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one transform operation is performed for each group, so that the component filters in each group share the transformed output from the respective transform operation; and wherein a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a

transform of the same length, and wherein only one inverse transform operation is performed for each group, so that the output from the component filters in each group is summed before being passed to the respective inverse transform operation.

In a further broad form of the invention there is provided a method for building a digital filter with a long impulse response and low latency, built by operating a number of smaller component filters in parallel and combining their outputs by addition, with each component filter operating with a different delay such that the net operation of the ensemble of said component filters is the same as a single filter with a longer impulse response, and the latency of the ensemble is equal to the shortest latency of the said component filters.

Preferably the component filters are implemented in different ways, with some filters adapted to provide low latency, and other filters adapted to provide longer filter lengths, such that the ensemble filter provides both low latency and long impulse response characteristics.

Preferably one or more of the component filters is implemented as a time-domain finite impulse response filter (built with multiply and add operations) and the remainder are implemented using a fast convolution method, such that the time-domain filter(s) provides the lowest latency portion of the ensemble impulse response, and the fast-convolution filter(s) provide the longer filter components.

Preferably the fast-convolution filters are built using the Discrete Fourier Transform or the Fast Fourier Transform.

Preferably the fast-convolution filters are built using the Modified Discrete Fourier Transform as described in this specification.

Preferably a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one transform operation is performed for each group, so that the component filters in each group share the transformed output from the respective transform operation.

In an alternative preferred form a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one inverse transform operation is performed for each group, so that the output from the component filters in each group is summed before being passed to the respective inverse transform operation.

In a further alternative preferred form a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one transform operation is performed for each group, so that the component filters in each group share the transformed output from the respective transform operation: wherein a group or groups of more than one of the

component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one inverse transform operation is performed for each group, so that the output from the component filters in each group is summed before being passed to the respective inverse transform operation.

In yet a further broad form of the invention there is provided a digital filter for filtering overlapped groupings of consecutive samples a, b, c, d....., said filter comprising a transform processor of length m, and N filter processors, each of length k, an adder adapted to sum the outputs as they are fed in parallel from said N filter processors, an inverse transform processor of length m: said consecutive samples a, b, c, d, ... being fed in blocks of length m, each of said blocks being passed through said transform processor and then through each of said N filter processors with a delay of zero before being passed through a first filter processor of said N filter processors, a delay of k before being passed through a second filter processor and so on up to a delay of (N-1)k before being passed through the Nth filter processor; whereby a filter of effective length Nk is effected with a latency corresponding to that of a conventional filter of length k.

In a further broad form of the invention there is provided a method of implementing a filter with a relatively high length/latency efficiency K; said method comprising transforming progressive, consecutive and overlapping

portions of input data into the frequency domain, to produce corresponding transformed data, storing said corresponding transformed data, so as to effect passing said portions of transformed data through a transform processor consecutively; feeding a first transformed portion through a first filter function processor; then feeding said first transformed portion through an N-1th filter function processor whilst an N-1th transformed portion is being fed through said first filter function processor; and so on so that a continuously moving N consecutive blocks of input data are transformed and passed through N filter function processors; adding the output from said N filter function processors; inverse transforming said output; performing a discard operation as necessary whereby consecutive portions of filtered output data are produced from said progressive, consecutive and overlapping portions of input data.

In a further broad form of the invention there is provided a method of implementing a filter with a relatively high length/latency efficiency K; said method comprising applying a mathematical transform to progressive, consecutive and overlapping portions of input data so as to produce corresponding transformed data; performing a mathematical operation on individual ones (i.e. two or more) of said transformed data; superposing (by addition) the data resulting from said mathematical operations so as to produce resultant data; applying an inverse mathematical transform to said resultant data so as to produce filtered output data.

Preferably each said filter function processor is processing a filter whose Impulse Response is half the length of the transform processor.

Preferably said transform processor is a Fast Fourier Transform Processor and said inverse transform processor is an inverse Fast Fourier Transform Processor and said filter function processor is effected by a multiply operation of transformed input data with an impulse response corresponding to a selected portion of a desired impulse response for the entire filter.

Preferably said method is optimized whereby an approximately equal amount of processing time is spent on Fourier transformation as on filter function (multiply) operation.

In a further broad form of the invention there is provided a method of implementing a filter with a relatively high length/latency efficiency K; said method comprising transforming progressive, consecutive and overlapping portions of input data into the frequency domain, performing a mathematical operation on individual ones of said transformed signals, superposing (by addition) the consecutive signals resulting from said mathematical operations, inverse transforming the resultant signal from the frequency domain back to the time domain and outputting said signal.

Preferably said transform is a Fast Fourier Transform and said inverse transform is an Inverse Fast Fourier Transform.

Preferably said mathematical operation is a vector multiply of the Fourier transformed input signal segment with the frequency response of the desired (time domain) filter characteristic.

Preferably the underlying operation is an overlap and discard operation on successive, overlapping portions of input data.

Alternately the underlying operation is an overlap and add operation on successive, overlapping portions of input dated.

In a particular preferred form the Fourier transform of any given overlapping block of input samples is taken only once and reused as required.

Preferably, the inverse Fourier transform of any given summed grouping of filtered data is performed only once.

In a particular preferred form the above described method is utilised to implement at least some of the filter modules of a composite electrical filter; said composite electrical filter comprising a plurality of filter modules arranged to receive in parallel an incoming input signal for filtering; the output from each of said filter modules being summed to produce a composite filtered output signal; each of said filter modules adapted to have an impulse response that is a selected portion of the impulse response of the composite filter.

It is preferable to minimise the overlap of the selected portions of impulse response, or to make them not overlap at all.

Preferably the length of each of said filter modules is different with the characteristic of the shorter length filter modules adapted to process first (or earlier) portions of said impulse response and longer length filter modules adapted to process following (or later) portions of said impulse response.

Preferably said longer length filter modules are adapted to filter progressively longer portions of said impulse response.

In a particular preferred form, the shortest module of said plurality of filter modules is a time-domain (low latency) filter whilst additional ones of said filter modules are longer fast-convolution (longer latency) filters implemented using the Fast Fourier Transform method described above or other traditional fast convolution methods.

It is an object of at least a further preferred embodiment of the present invention to provide a method and apparatus for performing relatively long convolutions on digital sampled data so as to provide relatively lower latency than is ordinarily incurred with other methods.

It is assumed that the filter characteristics can be modelled as approximately linear so that the principles of superposition can be applied.

Accordingly, in yet a further broad form of the invention, there is provided a composite electrical filter comprising a plurality of filter modules arranged to receive in parallel an incoming input signal for filtering; the output from each of said filter modules being summed to

produce a composite filtered output signal; each of said filter modules adapted to have an impulse response that is a selected portion of the impulse response of the composite filter.

It is preferable to minimise the overlap of the selected portions of impulse response, or to make them not overlap at all.

Preferably the length of each of said filter modules is different with the characteristic of the shorter length filter modules adapted to process first (or earlier) portions of said impulse response and longer length filter modules adapted to process following (or later) portions of said impulse response.

Preferably said longer length filter modules are adapted to filter progressively longer portions of said impulse response.

In a particular preferred form, the shortest module of said plurality of filter modules is a time-domain (low latency) filter whilst additional ones of said filter modules are longer fast-convolution (longer latency) filters.

Preferably only the shortest of said filter modules is a time domain filter.

In a further particular preferred form, where the number of said filter modules is N comprising filters F_1, $F_2 \ldots F_N$ filter module F_1 is a filter with very low latency implemented with time domain techniques whilst all other filter modules F_i are implemented with fast convolution techniques and these fast convolution filters

F_i are composed of a sequence of filters each with longer filter length than its predecessor and hence each with longer latency, but still preserving the property that $d_{i+1}=d_i+l_i$ whereby it is ensured that the composite filter F output formed by summing together the N component filter outputs has an impulse response without any "holes" in it.

In a further broad form of the invention there is provided a method of filtering sampled data so as to achieve a relatively long length but short latency filtering of said data, said method comprising passing said data in parallel through a plurality of sub-filters and summing the output samples from all of said sub-filters to produce filtered sampled data; and wherein the Impulse Response of each of said sub-filters is a selected portion of the desired Impulse Response of the filter characteristic required to produce said filtered sampled data from said sampled data and wherein each said selected portion is selected for each of said sub-filters such that the output from all of said sub-filters, when summed, behaves as if filtered through a filter having said desired Impulse Response.

In a particular implementation of the invention, there is provided a method and filter incorporating a fourier transform processor adapted to efficiently transform strings of real numbers; said processor operating according to the following method:

1. Take the input vector $x(k)$ of length N, where each element of $x(k)$ is real.

2. Create the vector $x'(l)$, a complex vector of length N/2 by application of the following:

$$x'(l)=[x(l)-jx(l+\frac{N}{2})]e^{\frac{-\pi jl}{N}} \quad for\ 0 \le l < N/2 \qquad (9)$$

3. Compute the N/2 point DFT of $x'(l)$ to give the N/2 complex result vector Z(p):

$$Z(p)=DFT_{(N/2)}[x'(l)] \qquad (10)$$

Embodiments of the invention will now be described with reference to the accompanying drawings wherein:-

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a generalised block diagram of a filter operation in the frequency domain,

Fig. 2 defines the basic terminology used for a convolution filter,

Fig. 3 defines the latency and length of the filter of Fig. 2,

Fig. 4 illustrates in a diagrammatic flow chart form, a prior art method of processing sampled data by a Fast Fourier Transform approach,

Fig. 5 further illustrates the approach of Fig. 4,

Fig. 6,7,8 develop a method of filtering according to a generalised first embodiment of the invention whereby a relatively high efficiency factor K can be achieved as compared with the approach of Fig. 4,

Fig. 9 is a diagram of an embodiment of the invention implementing the method of Fig. 8 where the number of sub-filters is 6,

Fig. 10 illustrates in block diagram form a summed filter a part of which can be implemented advantageously with the filter of Fig. 8,

Fig. 11 is a block diagram of the summed filter of Fig. 10 incorporating sub-filters some of which implement the method of Fig. 8,

Fig. 12 illustrates the manner of processing of an input signal by an example of the filter arrangement of Fig. 10 which utilised five filter portions,

Fig. 13 illustrates the manner of selection of the filter characteristics of the filter of Fig. 12,

Fig. 14 is a bloc diagram of an alternative implementation of the summed filter of Fig. 8,

Fig. 15 illustrates a typical flow of a (prior art) fast convolution algorithm implementation suitable for filters F_2 - F_5 of Fig. 12,

Fig. 16 illustrates a DFT engine which forms the basis for an explanation of a fourier transform algorithm optimised to process real number strings, and

Fig. 17 is a block diagram of a further embodiment of the invention wherein the summed filter of Fig. 10 is implemented utilizing a Modified Discrete Fourier Transform.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

1. High Efficiency Filter

Figure 4 illustrates the time-flow of a typical fast-convolution algorithm. This is an overlap-discard algorithm implemented using the Fast Fourier Transform (FFT). 2M words of input data that arrives during time segments a and b is processed fully during time segment c with a forward FFT, a vector multiply, and an inverse FFT. The resulting M words of output data are buffered, to be presented at the output during time segment d. The FFT and inverse FFT(IFFT) are only used to transform the data between the time-domain and frequency domain. The actual filter operation is executed in the vector multiply operation, which actually takes only a small fraction of the total compute time. So, the relevant parameters of the filter of Fig. 4 are:-

Length = M,

Latency = 2M,

and, therefore K = 0.5

With reference to Figs. 6, 7 and 8, the rationale behind the method and apparatus according to at least one embodiment of the present invention is derived.

Fig. 6 illustrates a filter of length ML where the filter characteristics of each of the component filters F1,

F2, ... F_n are separate, discrete, component portions of the desired filter characteristic for the entire filter assembly. The delays L, 2L.... (M-1) L are imposed as illustrated so as to recreate, following addition, an output y_k equivalent to that achieved by passing input samples x_k through a filter having the filter characteristic from which the filter characteristic portions for filters F1, F2..... were derived. Fig. 7 is derived by implementing the filters F1, F2.... of Fig. 6 using the Fast Fourier Transform algorithm of Fig. 5.

With reference to Fig. 8, reorganisation of the filter of Fig. 7 allows the use of only one Fast Fourier Transform module 11 and one inverse Fast Fourier Transform module 12. It is implicit that the Fast Fourier Transform module is adapted to process a block of samples from input x_k equal to twice the length of each of the filters Filter 1, Filter 2, Filter 3...... illustrated in Fig. 8.

As previously stated the filter characteristic (impulse response) of each consecutive filter F1, F2.... is taken from and corresponds to consecutive corresponding portions from the impulse response desired of the entire filter module.

The time delay L before each Fast Fourier transformed block of data is passed through the next filter is equal to half the sample length originally processed by the Fast Fourier Transform module.

Figure 9 shows the computation of one block of output data, in a similar style to Figure 4, but using the improved length/latency efficiency method derived in Figs. 6, 7 and 8.

The method of Fig. 8 as used by Fig. 9 is summarised below.

During time segment h, the input data that arrived during time segments f and g is FFT'd and the resulting block of Frequency Domain Input Data is stored for future use. We then compute the next block of Frequency Output Data, which is inverse FFT'd and presented as output during time segment i. The old way of computing fast-convolution simply took the latest block of Frequency Domain Input Data, and multiplied it by a vector that represents the desired filter response, to get the new Frequency Domain Output Data. The improved length/latency efficiency method uses a number of previous Frequency Domain Input Data blocks to compute the new Frequency Domain Output Data block, as shown in Figure 9. In this example, the blocks of filter data are called Filter A, Filter B, ..., Filter F. In this example, the filter implemented is 6 times as long as the filter implemented in Figure 4, but with no greater latency. By comparison with Fig. 4, the relevant parameters of the filter of Fig. 9 are:-

Length = 6M,

Latency = 2M,

and, therefore K = 3

Fig. 8 summarises the logic behind the implementation of the embodiment of Fig. 9.

Particularly, it will be noted that the progressive delays L, 2L, 3L, ...(M-1)L of Fig. 8 are achieved in Fig. 9 by the taking of delayed, overlapped groupings of consecutive samples a, b, c, d, ...

The above described filter arrangement can be used advantageously in a low-latency FIR filter arrangement such as illustrated in Fig. 10.

Figure 10 shows an architecture for implementing an FIR filter by adding together N filters. If each filter is characterised as: Filter F_i, latency d_i, length l_i, then generally the N filters are chosen so that their latencies are ordered in ascending order, and furthermore $d_{i+1}=d_i+l_i$. This means that the first non-zero value in the impulse response of filter F_{i+1}, comes immediately following the last non-zero value in the impulse response of filter F_i. Hence this summation of filters results in a single, longer filter with its impulse response being the sum of the impulse responses of the N component filters.

The important property of this filter is the length/latency efficiency, K, which is higher than any of the component filter efficiencies.

That is, the filter of Fig. 10 uses the technique of adding together several filters to form a new filter which is as long as the sum of the component filter lengths, and whose latency is as short as the latency of the lowest-latency component filter.

Fig. 11 shows an implementation of the low-latency filter 10 of Fig. 10 wherein there are three filter modules F1, F2, F3. The first module F1 is a low-latency (d=0) time domain filter whilst filters F2 and F3 are implemented according to the embodiment described in respect of Figs. 8 and 9.

2. A low-latency FIR filter

As previously described. Fig. 10 shows an architecture for implementing an FIR filter by adding together N filters. If each filter is characterised as: Filter F_i, latency d_i, length l_i, then generally the N filters are chosen so that their latencies are ordered in ascending order, and furthermore $d_{i+1}=d_i+l_i$. This means that the first non-zero value in the impulse response of filter F_{i+1}, comes immediately following the last non-zero value in the impulse response of filter F_i. Hence this summation of filters results in a single, longer filter with its impulse response being the sum of the impulse responses of the N component filters.

The important property of this filter is the length/latency efficiency, K, which is higher than any of the component filter efficiencies.

That is, the filter of Figs. 10 and 12 uses the technique of adding together several filters to form a new filter which is as long as the sum of the component filter lengths, and whose latency is as short as the latency of the lowest-latency component filter.

Particularly, the composite filter assembly of Fig. 12 utilises the technique of combining a first time-domain (low latency) filter with additional fast-convolution (longer latency) filters to maximise filter length while minimising latency. This technique is implemented by adding together N filters, F_1, F_2, ...F_N where F_1 is a filter with very low latency, implemented with time-domain techniques, and the

other filters. F_i, are each implemented with fast-convolution techniques. More specifically, the assembly adopts the technique whereby the N-1 fast-convolution filters, Fi, are composed of a sequence of filters, each with longer filter length than its predecessor, and hence each with longer latency, but still preserving the property that $d_{i+1}=d_i+l_i$. This ensures that the filter, F, which is made by summing together the N component filters, has an impulse response without any "holes" in it.

With particular reference to Fig. 12 the composite filter F comprises five filter portions F1, F2, F3, F4 and F5. The impulse response a_k of the composite filter F is illustrated at the top of Fig. 12 and has a total sample length of 1024 samples.

Filter F1 has an impulse response comprising the first 64 samples of the impulse response a_k. That is, the filter has a length of 64 samples. The filter as implemented has a low latency filter (such as is referenced in Motorola document APR 7/D in respect of the DSP 56000 series of Integrated Circuits). This filter has an effective latency of 0.

The subsequent filters F2, F3, F4, F5 are implemented using fast convolution digital techniques. Fig. 15 illustrates the basic algorithm for such techniques which comprises taking the fast Fourier transform of the incoming sampled data, multiplying the transformed data samples by the impulse response of the filter, converting the fast Fourier transformed data samples back to the time domain by use of an

inverse fast Fourier transform and then outputting the data. An overlap/discard method is used whereby only a portion of the output data is utilised.

The length and latency of the additional filters F2, F3, F4, F5 is selected according to the rule illustrated diagrammatically in Fig. 13, whereby each filter portion has a latency equal to the sum of the length and the latency of the immediately preceding filter portion.

In this case the end result is a filter having a total length of 1024 samples and a latency of 0.

Fig. 14 illustrates a variation of the filter of Fig. 8 wherein delay is introduced after the filter algorithm is applied in the frequency domain.

3. Optimized Real String Handling

With reference to Fig. 16, a common method of frequency analysis is via the Discrete Fourier Transform (hereafter referred to as the DFT), which can be implemented efficiently in electronic apparatus using the Fast Fourier Transform algorithm (hereafter referred to as the FFT).

The DFT is formulated to operate efficiently when its input data and output data are both complex (having a real and imaginary component). When the data input to the DFT is real, the output data from the operation will contain some redundancy, indicating that some of the processing that led to this output data was unnecessary.

In this embodiment what is described is a new transform for operating on real numbers in the digital environment, that has many of the same applications as the DFT, but

without the inefficiencies of the DFT for operation on real numbers. For the purposes of this document, the algorithm described herein will be named the Modified Discrete Fourier Transform (MDFT).

The DFT is computed according to the equation below:

$$X(n)=\sum_{k=0}^{N-1} x(k)e^{\frac{-2\pi jnk}{N}} \qquad (1)$$

If the input data x(k) is real (ie. it has no imaginary component), the output data X(n) has the following properties:

$$\begin{aligned} &X(0) \in \Re \\ &X(\tfrac{N}{2}) \in \Re \\ &X(n) = [X(N-n)]^{*} \; for \; 0<n<\tfrac{N}{2} \end{aligned} \qquad (2)$$

Where the * operator is used to signify complex conjugation. This means that the imaginary part of X(0), the imaginary part of X(N/2) and all {X(n):N/2<n<N} are redundant. The process of extracting only the necessary information out of the DFT output is therefore not trivial.

An alternative transform is shown below:

$$Y(n) = \sum_{k=0}^{N-1} x(k)e^{\frac{-2\pi jk(n+\frac{1}{2})}{N}} \qquad (3)$$

This is like a standard DFT except that the output vector Y(n) represents the signal's frequency components at different frequencies to the DFT. The output vector Y(n) has redundancies (just as the DFT output X(n) has), except that

the redundant part of the data is more clearly extracted from Y(n) that from X(n). The redundancy in Y(n) that results when x(k) is real can be expressed as follows:

$$Y(n) = [Y(N-1-n)]^* \qquad (4)$$

This implies that the second half of this vector Y(n) is simply the complex conjugate of the first half, so that only the first half of the output vector is required to contain all of the information, when x(k) is real.

An alternative view of the above equation is that all of the odd elements of the vector are simply the complex conjugate of the even elements:

$$\begin{aligned} Y(1) &= [Y(N-2)]^* \\ Y(3) &= [Y(N-4)]^* \\ &\cdots \\ Y(N-3) &= [Y(2)]^* \\ Y(N-1) &= [Y(0)]^* \end{aligned} \qquad (5)$$

This means that we only need to compute the even elements of Y(n) to obtain all of the required information from the modified DFT of the real signal x(k). We can name the array Z(p) the array that contains the even elements from Y(n), as follows:

$$Z(p) = Y(2p) \textit{ for } 0 \le p < \frac{N}{2} \qquad (6)$$

Based on our previous equation for Y(n), we get:

$$Z(p) = Y(2p) = \sum_{k=0}^{N-1} x(k) e^{\frac{-2\pi kj(2p+\frac{1}{2})}{N}} \qquad (7)$$

which after some manipulation becomes:

$$Z(p) = \sum_{k=0}^{\frac{N}{2}-1} [x(k) - jx(k+\tfrac{N}{2})] e^{\frac{-\pi jk}{N}} e^{\frac{-2\pi jpk}{(N/2)}} \qquad (8)$$

if we create an N/2 length complex vector from the N length real vector:

$$x'(l) = [x(l) - jx(l+\tfrac{N}{2})]e^{\frac{-\pi jl}{N}} \quad for\ 0 \le l < N/2 \qquad (9)$$

then we can see that:

$$Z(p) = DFT_{(N/2)}[x'(l)] \qquad (10)$$

This means that we have computed the vector Z(p) by using a DFT of length N/2.

We say that Z(p)=MDFT[x(k)] (where MDFT indicates the Modified Discrete Fourier Transform operator). The procedure to follow for computing Z(p) is then as follows:

1. Take the input vector x(k) of length N, where each element of x(k) is real.
2. Create the vector x'(l), a complex vector of length N/2 by the method of equation (9) above.
3. Compute the N/2 point DFT of x'(l) to give the N/2 complex result vector Z(p).

The MDFT has many properties that make it useful in similar applications to the DFT. Firstly, it can be used to perform linear convolution in the same way as the DFT. Secondly, it has an inverse transform that looks very similar to the forward transform:

$$x'(k) = IDFT_{(N/2)}[Z(p)] \qquad (11)$$

where IDFT indicates the N/2 point Inverse Discrete Fourier

Transform.

The algorithm can be implemented in an electronic apparatus supplied with a set of N real numbers and producing N/2 complex output numbers, representing the MDFT of the input data. This apparatus uses digital arithmetic elements to perform each of the arithmetic operations as described in the preceding text.

Another embodiment of the present invention is a pair of apparatus, the first of which computes an MDFT as described in the previous paragraph, and the second of which computes an inverse MDFT, using the arithmetic procedures described previously in this document. By passing overlapped blocks of data from a continuous stream of input data through the MDFT computer, then multiplying the Z(p) coefficients by appropriate filter coefficients, then passing the resulting data through the Inverse MDFT computer, and recombining segments of output data appropriately, a modified Fast Convolution processor can be built.

The above described a modification to the DFT that makes it more useful in a number of applications particularly but not limited to the real time filter applications previously described. All of these extensions to the DFT can also be applied to the FFT algorithm and other fast implementations of the DFT.

<u>Example 1</u>

Fig. 17 illustrates an implementation of the summed filter of Fig. 11 wherein the Modified Discrete Fourier Transform (MDFT) described immediately above is applied for

the purposes of transforming the data stream into the frequency domain and the corresponding Inverse Modified Discrete Fourier Transform (IMDFT) is applied following application of the filter algorithm and prior to discard for conversion from the frequency domain.

In filter F2 of Fig. 17, the MDFT takes 64 real words of input and produces 32 complex words of output. The IMDFT takes 32 complex words of input and produces 64 real words of output.

In filter F3 of Fig. 17 the MDFT takes 256 real words of input and produces 128 complex words of output. The IMDFT takes 128 complex words of input and produces 256 real words of output.

The filter of Fig. 17 is implemented using a Motorola DSP 56001 processor incoporating software (bootable from ROM or from another host computer) to implement the algorithm. The delay elements are implemented using a bank of external memory chips comprising three MCM 6206 memory chips.

Data input and output between the analog and digital domains is effected by an ADC and DAC chip, the Crystal CS 4216, communicating via the syncrhonous serial communication port of the DSP 56001.

INDUSTRIAL APPLICABILITY

Embodiments of the invention may be applied to digital filters implemented in software, hardware or a combination of both for applications such as audio filtering or electronic modelling of acoustic system characteristics. The method is broadly applicable in the field of signal processing and can

be used to advantage, for example. in: adaptive filtering; audio reverberation processing; adaptive echo cancellation; spatial processing; virtual reality audio; correlation, radar; radar pulse compression; deconvolution; seismic analysis; telecommunications; pattern recognition; robotics; 3D acoustic modelling; audio post production (including oralisation, auto reverberant matching); audio equalisation; compression; sonar; ultrasonics; secure communication systems; digital audio broadcast, acoustic analysis; surveillance; noise cancellation; echo cancellation.

The above describes only some embodiments of the present invention and modifications obvious to those skilled in the art can be made thereto without departing from the scope and spirit of the present invention.

CLAIMS

1. A digital filter with a long impulse response and low latency, built by operating a number of smaller component filters in parallel and combining their outputs by addition, with each component filter operating with a different delay such that the net operation of the ensemble of said component filters is the same as a single filter with a longer impulse response, and the latency of the ensemble is equal to the shortest latency of the said component filters.

2. The filter of claim 1 wherein the component filters are implemented in different ways, with some filters adapted to provide low latency, and other filters adapted to provide longer filter lengths, such that the ensemble filter provides both low latency and long impulse response characteristics.

3. The filter of claim 1 wherein one or more of the component filters is implemented as a time-domain finite impulse response filter (built with multiply and add operations) and the remainder are implemented using a fast convolution method, such that the time-domain filter(s) provides the lowest latency portion of the ensemble impulse response, and the fast-convolution filter(s) provide the longer filter components.

4. The filter of claim 3 wherein the fast-convolution filters are built using the Discrete Fourier Transform or the Fast Fourier Transform.

5. The filter of claim 3 wherein the fast-convolution filters are built using the Modified Discrete Fourier Transform as claimed in claim 33.

6. The filter of claim 1 wherein a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one transform operation is performed for each group, so that the component filters in each group share the transformed output from the respective transform operation.

7. The filter of claim 1 wherein a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one inverse transform operation is performed for each group, so that the output from the component filters in each group is summed before being passed to the respective inverse transform operation.

8. The filter of claim 1 wherein a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one transform operation is performed for each group, so that the component filters in each group share the transformed output from the respective transform operation; and wherein a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one inverse transform operation is performed for each group, so that the output from the

component filters in each group is summed before being passed to the respective inverse transform operation.

9. A method for building a digital filter with a long impulse response and low latency, built by operating a number of smaller component filters in parallel and combining their outputs by addition, with each component filter operating with a different delay such that the net operation of the ensemble of said component filters is the same as a single filter with a longer impulse response, and the latency of the ensemble is equal to the shortest latency of the said component filters.

10. The method of claim 9 wherein the component filters are implemented in different ways, with some filters adapted to provide low latency, and other filters adapted to provide longer filter lengths, such that the ensemble filter provides both low latency and long impulse response characteristics.

11. The method of claim 9 wherein one or more of the component filters is implemented as a time-domain finite impulse response filter (built with multiply and add operations) and the remainder are implemented using a fast convolution method, such that the time-domain filter(s) provides the lowest latency portion of the ensemble impulse response, and the fast-convolution filter(s) provide the longer filter components.

12. The method of claim 11 wherein the fast-convolution filters are built using the Discrete Fourier Transform or the Fast Fourier Transform.

13. The method of claim 11 wherein the fast-convolution filters are built using the Modified Discrete Fourier Transform as claimed in claim 33.

14. The method of claim 9 wherein a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one transform operation is performed for each group, so that the component filters in each group share the transformed output from the respective transform operation.

15. The method of claim 9 wherein a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one inverse transform operation is performed for each group, so that the output from the component filters in each group is summed before being passed to the respective inverse transform operation.

16. The method of claim 9 wherein a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same length, and wherein only one transform operation is performed for each group, so that the component filters in each group share the transformed output from the respective transform operation; wherein a group or groups of more than one of the component filters are implemented using fast convolution techniques with the component filters in each group using a transform of the same

length, and wherein only one inverse transform operation is performed for each group, so that the output from the component filters in each group is summed before being passed to the respective inverse transform operation.

17. A digital filter for filtering delayed, overlapped groupings of consecutive samples a, b, c, d..... said filter comprising a transform processor of length m, and N filter processors, each of length k, an adder adapted to sum the outputs as they are fed in parallel from said N filter processors, an inverse transform processor of length m: said consecutive samples a, b, c, d, ... being fed in blocks of length m, each of said blocks being passed through said transform processor and then through each of said N filter processors with a delay of zero before being passed through a first filter processor of said N filter processors, a delay of k before being passed through a second filter processor and so on up to a delay of (N-1)k before being passed through the Nth filter processor; whereby a filter of effective length Nk is effected with a latency corresponding to that of a conventional filter of length k.

18. The filter of claim 17 wherein said transform processor is a form of Fourier Transform Processor and said inverse transform processor is a form of inverse Fourier Transform Processor and said filter function processor is effected by a multiply operation of transformed input data with frequency response corresponding to a selected portion of a desired impulse response for the entire filter.

19. A method of implementing a filter with a relatively high length/latency efficiency K; said method comprising transforming progressive, consecutive and overlapping portions of input data into the frequency domain, performing a mathematical operation on individual ones of said transformed signals, superposing (by addition) the consecutive signals resulting from said mathematical operations, inverse transforming the resultant signal from the frequency domain back to the time domain and outputting said signal.

20. A method of implementing a filter with a relatively high length/latency efficiency K; said method comprising applying a mathematical transform to progressive, consecutive and overlapping portions of input data so as to produce corresponding transformed data; performing a mathematical operation on individual ones (i.e. two or more) of said transformed data; superposing (by addition) the data resulting from said mathematical operations so as to produce resultant data; applying an inverse mathematical transform to said resultant data so as to produce filtered output data.

21. The method of claim 20 wherein said transform is a form of Fourier Transform and said inverse transform is a form of inverse Fourier Transform and said mathematical operation is a vector multiply of the Fourier transformed input signal segment with the frequency responses of segments of the desired (time domain) filter characteristic.

22. The method of claim 21 wherein the underlying operation is an overlap operation on successive, overlapping portions

of input data and a discard or add operation on succession overlapping blocks of output data (commonly referred to as an overlap/save or overlap/add method).

23. The method of claim 20 wherein said method is utilised to implement at least some of the filter modules of a composite electrical filter; said composite electrical filter comprising a plurality of filter modules arranged to receive in parallel an incoming input signal for filtering; the output from each of said filter modules being summed to produce a composite filtered output signal; each of said filter modules adapted to have an impulse response that is a selected portion of the impulse response of the composite filter.

24. The method of claim 23 including the step of minimising the overlap of the selected portions of impulse response, or to make them not overlap at all.

25. The method of claim 24 wherein the length of each of said filter modules is different with the characteristic of the shorter length filter modules adapted to process first (or earlier) portions of said impulse response and longer length filter modules adapted to process following (or later) portions of said impulse response.

26. The method of claim 25 wherein the shortest module of said plurality of filter modules is a time-domain (low latency) filter whilst additional ones of said filter modules are longer fast-convolution (longer latency) filters.

27. A composite electrical filter comprising a plurality of filter modules arranged to receive in parallel an incoming

input signal for filtering; the output from each of said filter modules being summed to produce a composite filtered output signal; each of said filter modules adapted to have an impulse response that is a selected portion of the impulse response of the composite filter.

28. The filter of claim 27 adapted to minimise the overlap of the selected portions of impulse response, or to make them not overlap at all.

29. The filter of claim 28 wherein the length of each of said filter modules is different with the characteristic of the shorter length filter modules adapted to process first (or earlier) portions of said impulse response and longer length filter modules adapted to process following (or later) portions of said impulse response.

30. The filter of claim 29 wherein the shortest module of said plurality of filter modules is a time-domain (low latency) filter whilst additional ones of said filter modules are longer fast-convolution (longer latency) filters.

31. The filter of claim 27 where the number of said filter modules is N comprising filters F_1, $F_2 \ldots F_N$ and wherein filter module F_1 is a filter with very low latency implemented with time domain techniques whilst all other filter modules F_i are implemented with fast convolution techniques and these fast convolution filters F_i are composed of a sequence of filters each with longer filter length than its predecessor and hence each with longer latency, but still preserving the property that $d_{i+1}=d_i+l_i$ whereby it is ensured that the composite

filter F output formed by summing together the N component filter outputs has an impulse response without any "holes" in it.

32. A method of filtering sampled data so as to achieve a relatively long length but short latency filtering of said data, said method comprising passing said data in parallel through a plurality of sub-filters and summing the output samples from all of said sub-filters to produce filtered sampled data; and wherein the Impulse Response of each of said sub-filters is a selected portion of the desired Impulse Response of the filter characteristic required to produce said filtered sampled data from said sampled data and wherein each said selected portion is selected for each of said sub-filters such that the output from all of said sub-filters, when summed, behaves as if filtered through a filter having said desired Impulse Response.

33. A fourier transform processor adapted to efficiently transform strings of real numbers; said processor operating according to the following method:

1. Take the input vector x(k) of length N, where each element of x(k) is real.
2. Create the vector x'(l), a complex vector of length N/2 by application of the following:

$$x'(l) = [x(l) - jx(l+\frac{N}{2})]e^{\frac{-\pi jl}{N}} \quad for\ 0 \le l < N/2 \qquad (9)$$

3. Compute the N/2 point DFT of x'(l) to give the N/2 complex result vector Z(p):

$$Z(p) = DFT_{(N/2)}[x'(l)] \qquad (10)$$

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member					
US	4992967	EP	312463	FR	2622069		
EP	250048	AU	74466/87	CA	1266893	JP	63004710
		NL	8601604	US	4807173		
EP	448758	US	5159565				
WO	88/03341	EP	288577	US	4951269		
US	4623980	CA	1201178	DE	3118473	EP	65210
		JP	58030219				

END OF ANNEX

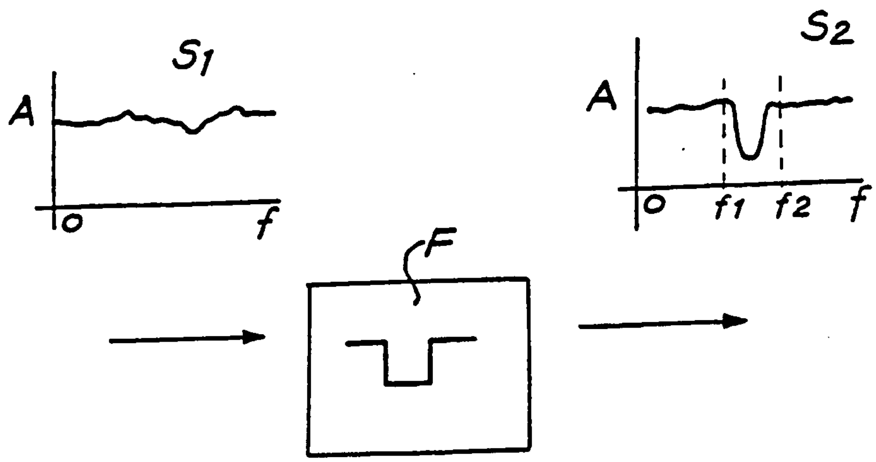


FIG. 1

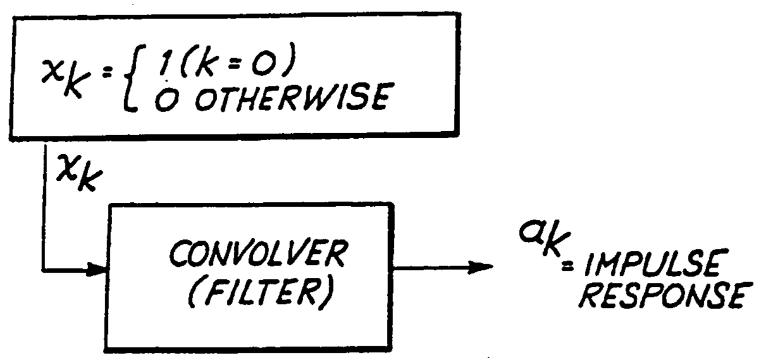

$x_k = \begin{cases} 1 (k=0) \\ 0 \text{ OTHERWISE} \end{cases}$
x_k
CONVOLVER (FILTER)
a_k = IMPULSE RESPONSE

FIG. 2

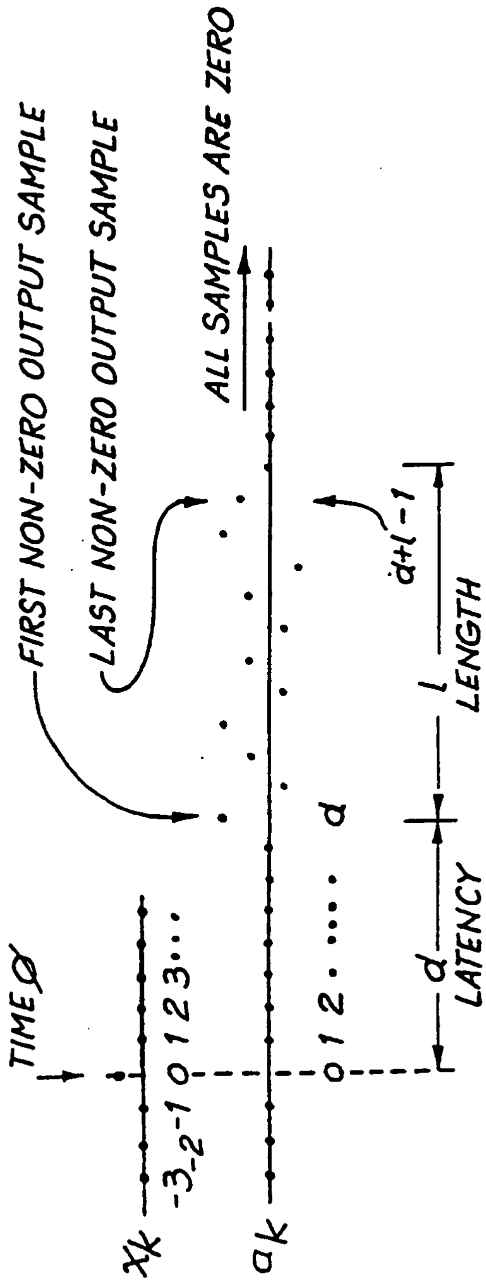

FIRST NON-ZERO OUTPUT SAMPLE
LAST NON-ZERO OUTPUT SAMPLE
ALL SAMPLES ARE ZERO
TIME Ø
x_k
a_k
-3 -2 -1 0 1 2 3 ...
0 1 2 .... d
d
LATENCY
l
LENGTH
d+l-1

FIG.3

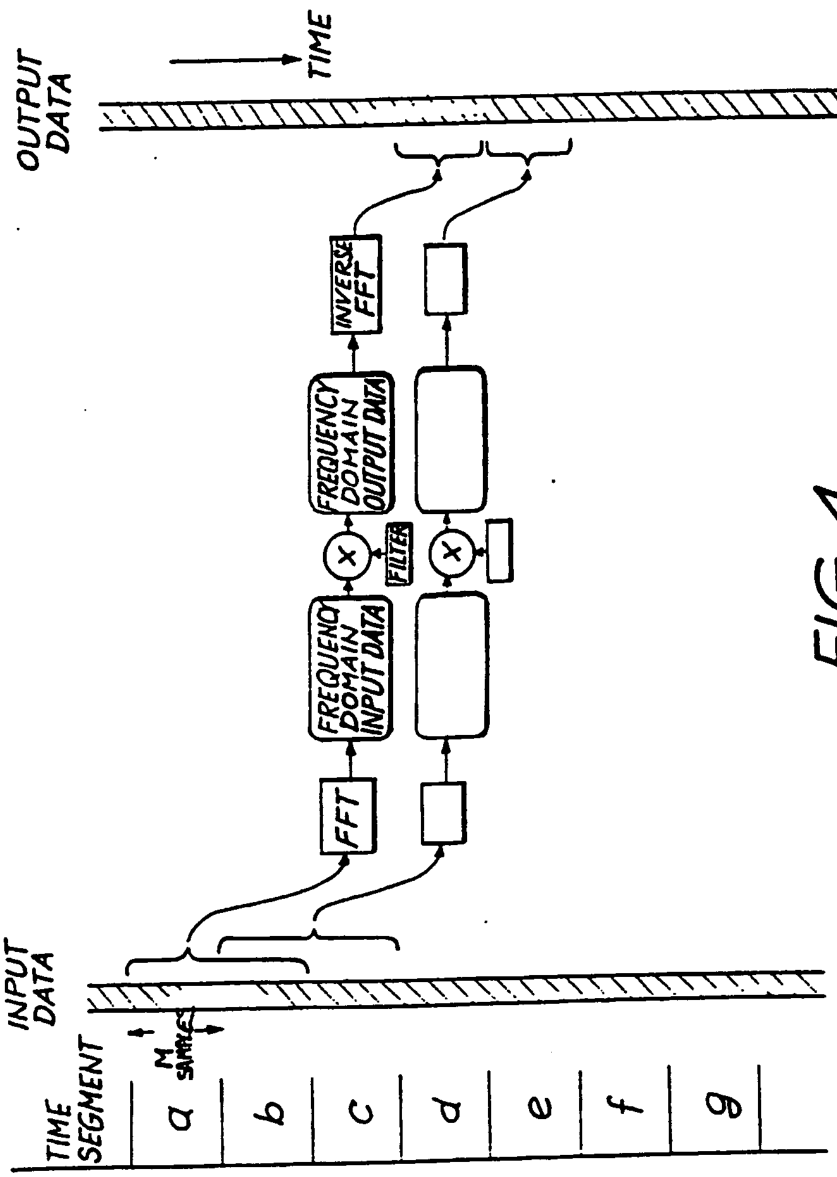
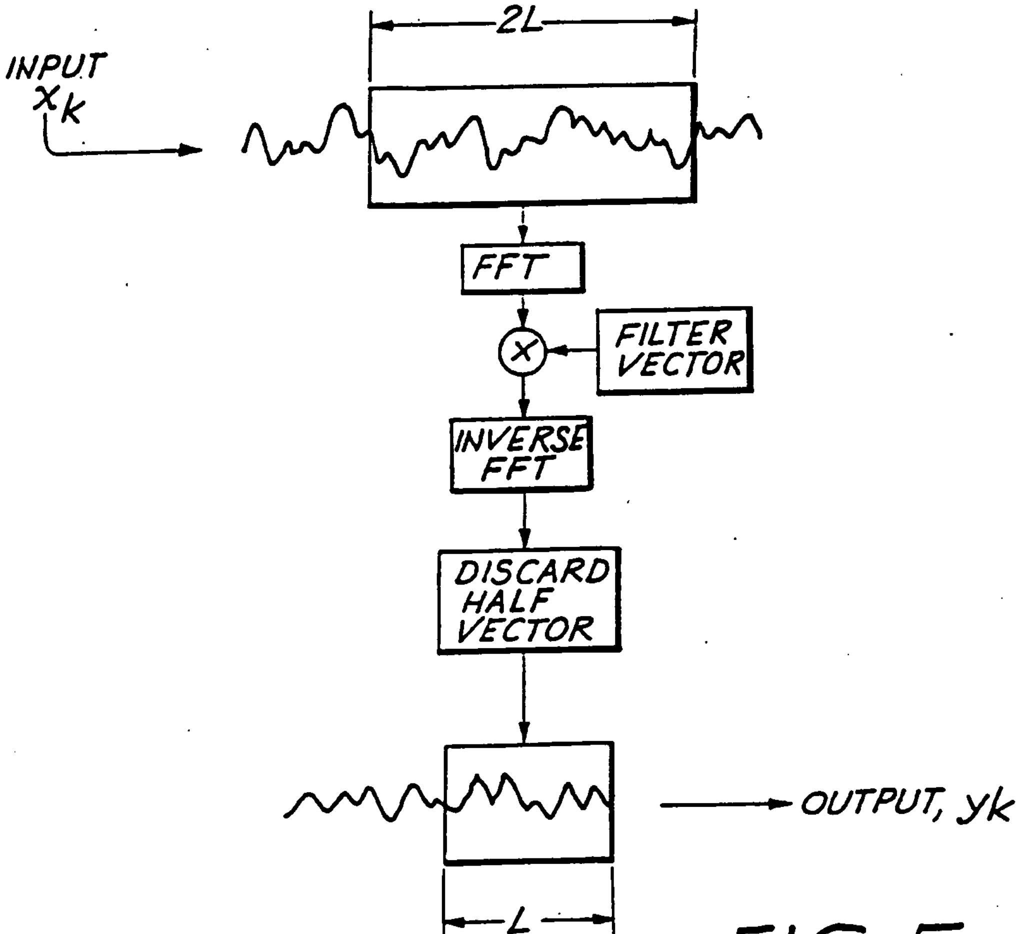

OUTPUT DATA
TIME
INVERSE FFT
FREQUENCY DOMAIN OUTPUT DATA
FILTER
X
FREQUENCY DOMAIN INPUT DATA
FFT
INPUT DATA
TIME SEGMENT
M SAMPLES
a
b
c
d
e
f
g

FIG. 4



2L
INPUT
x_k
FFT
FILTER VECTOR
INVERSE FFT
DISCARD HALF VECTOR
OUTPUT, yk
L

FIG. 5

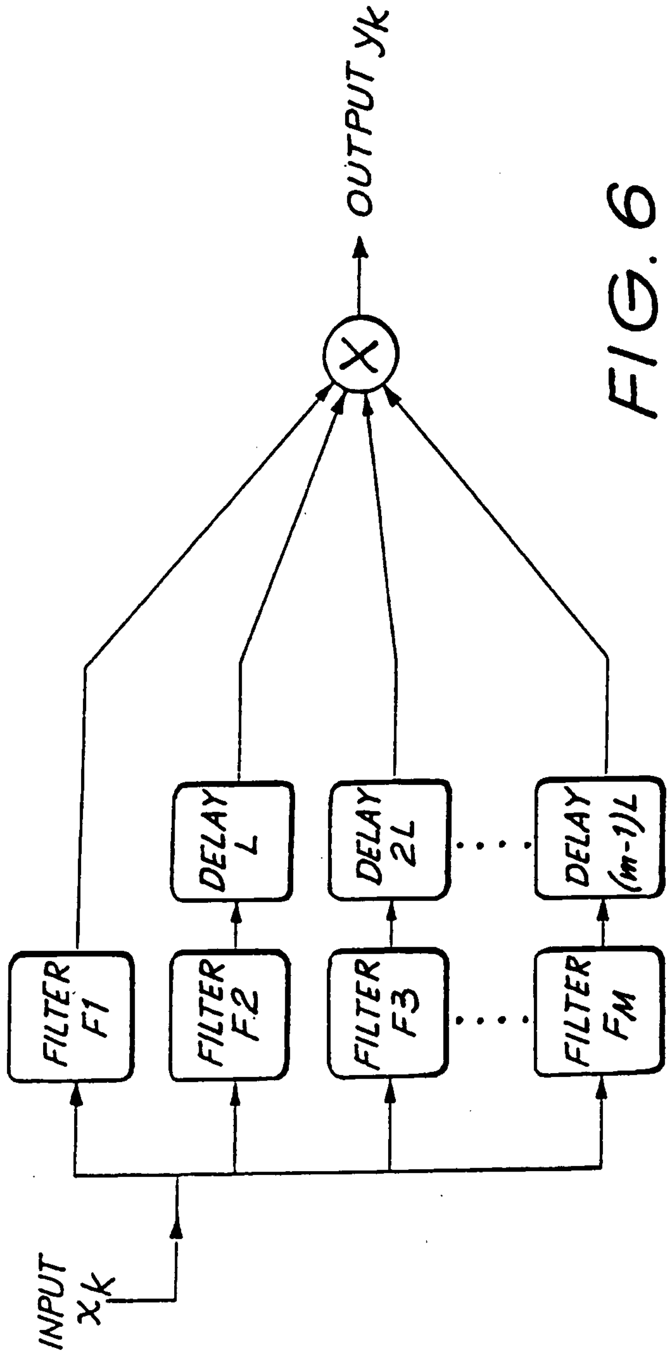

OUTPUT y_k
FILTER F1
FILTER F2
FILTER F3
FILTER FM
DELAY L
DELAY 2L
DELAY (m-1)L
INPUT x_k

FIG. 6

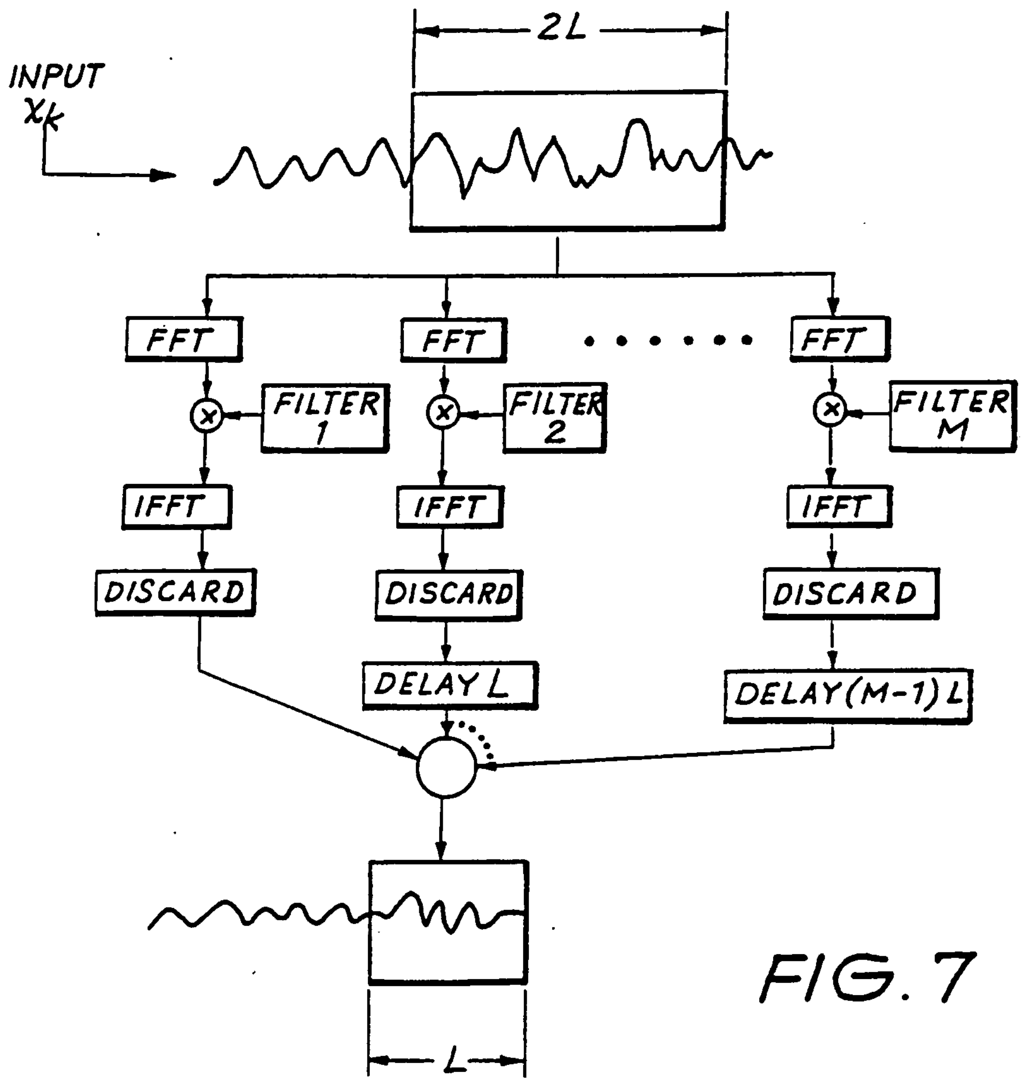

2L
INPUT
x_k
FFT
FFT
FFT
FILTER 1
FILTER 2
FILTER M
IFFT
IFFT
IFFT
DISCARD
DISCARD
DISCARD
DELAY L
DELAY (M-1) L
L

FIG. 7

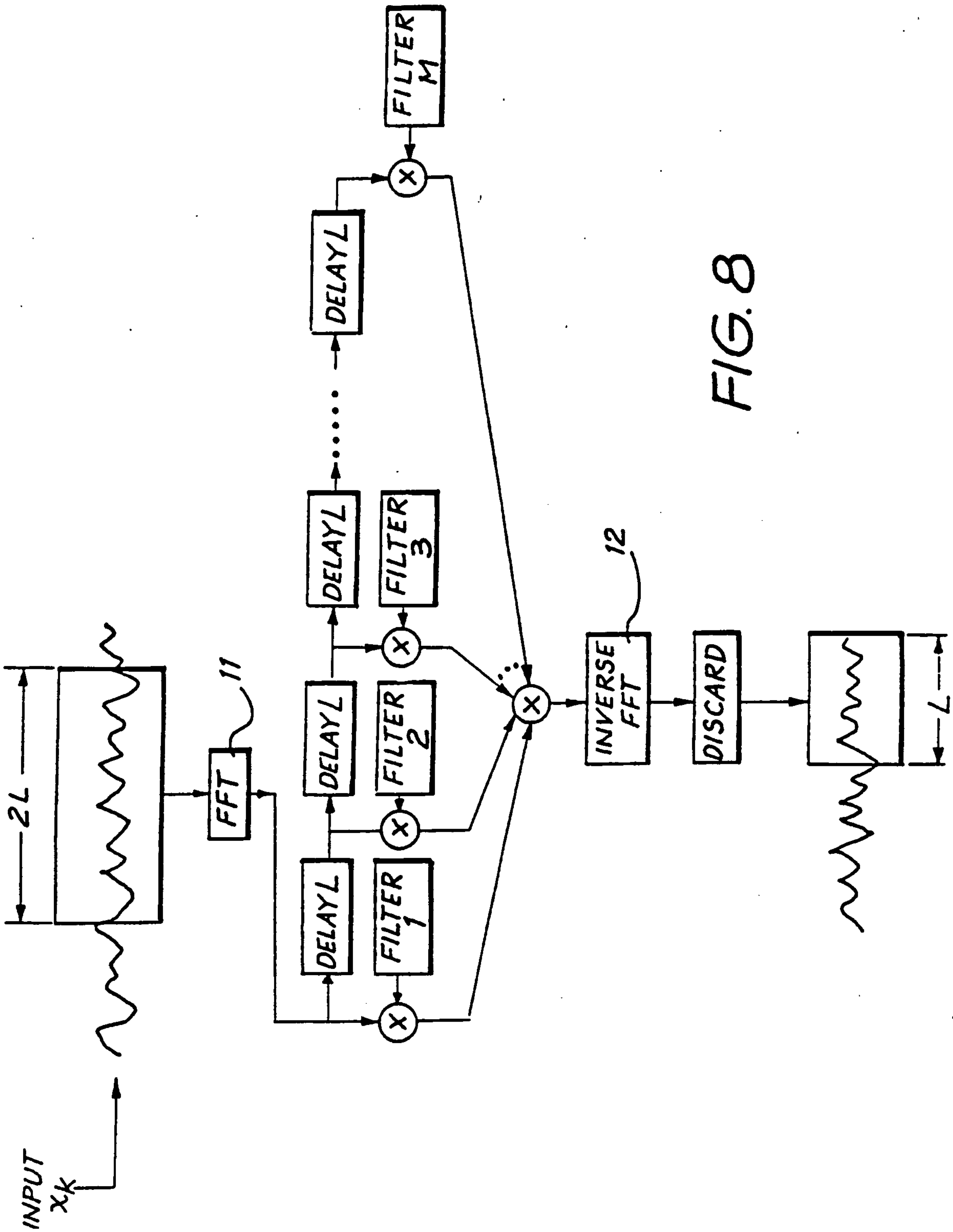

INPUT x_k
2L
FFT
11
DELAY L
DELAY L
DELAY L
DELAY L
FILTER 1
FILTER 2
FILTER 3
FILTER M
12
INVERSE FFT
DISCARD
L

FIG. 8

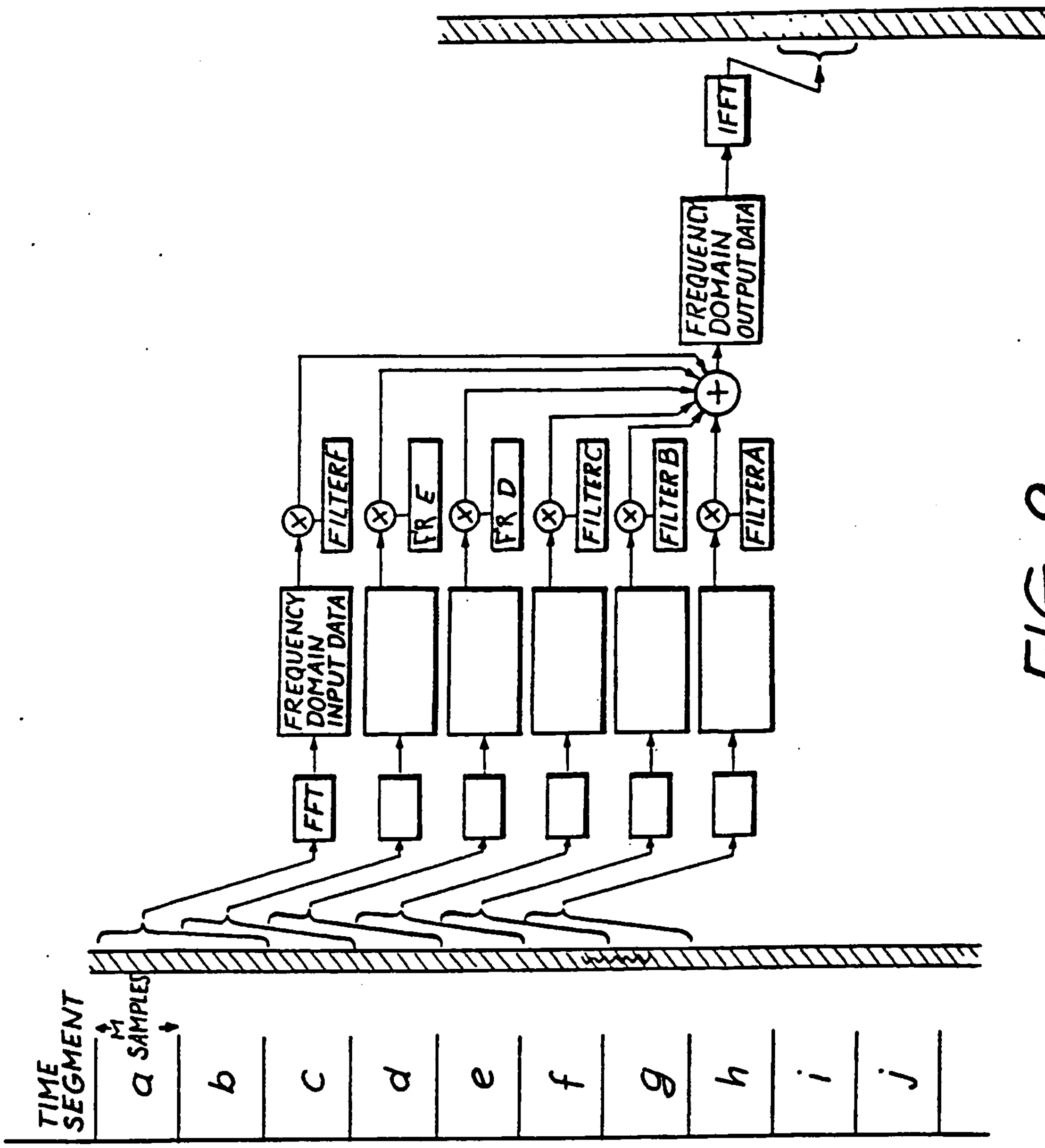

TIME SEGMENT
M SAMPLES
a
b
c
d
e
f
g
h
i
j
FFT
FREQUENCY DOMAIN INPUT DATA
FILTER F
FR E
FR D
FILTER C
FILTER B
FILTER A
FREQUENCY DOMAIN OUTPUT DATA
IFFT

FIG. 9

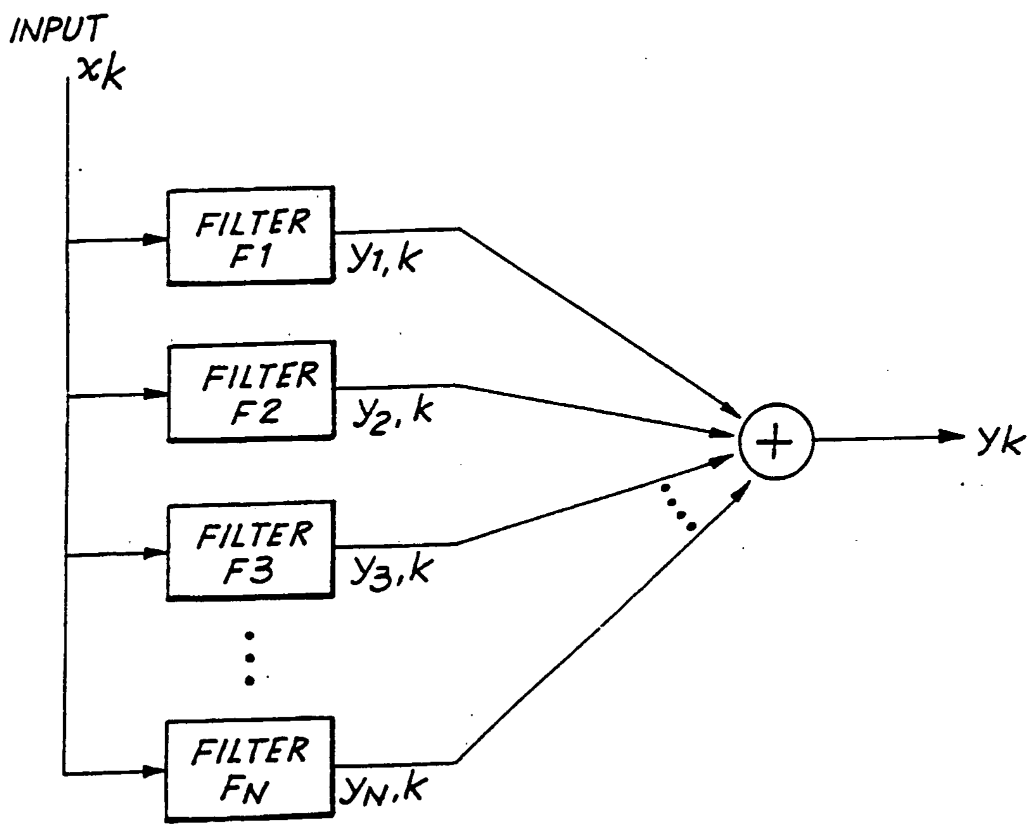

INPUT
x_k
FILTER F1
FILTER F2
FILTER F3
FILTER FN
$y_{1,k}$
$y_{2,k}$
$y_{3,k}$
$y_{N,k}$
+
y_k

FIG. 10

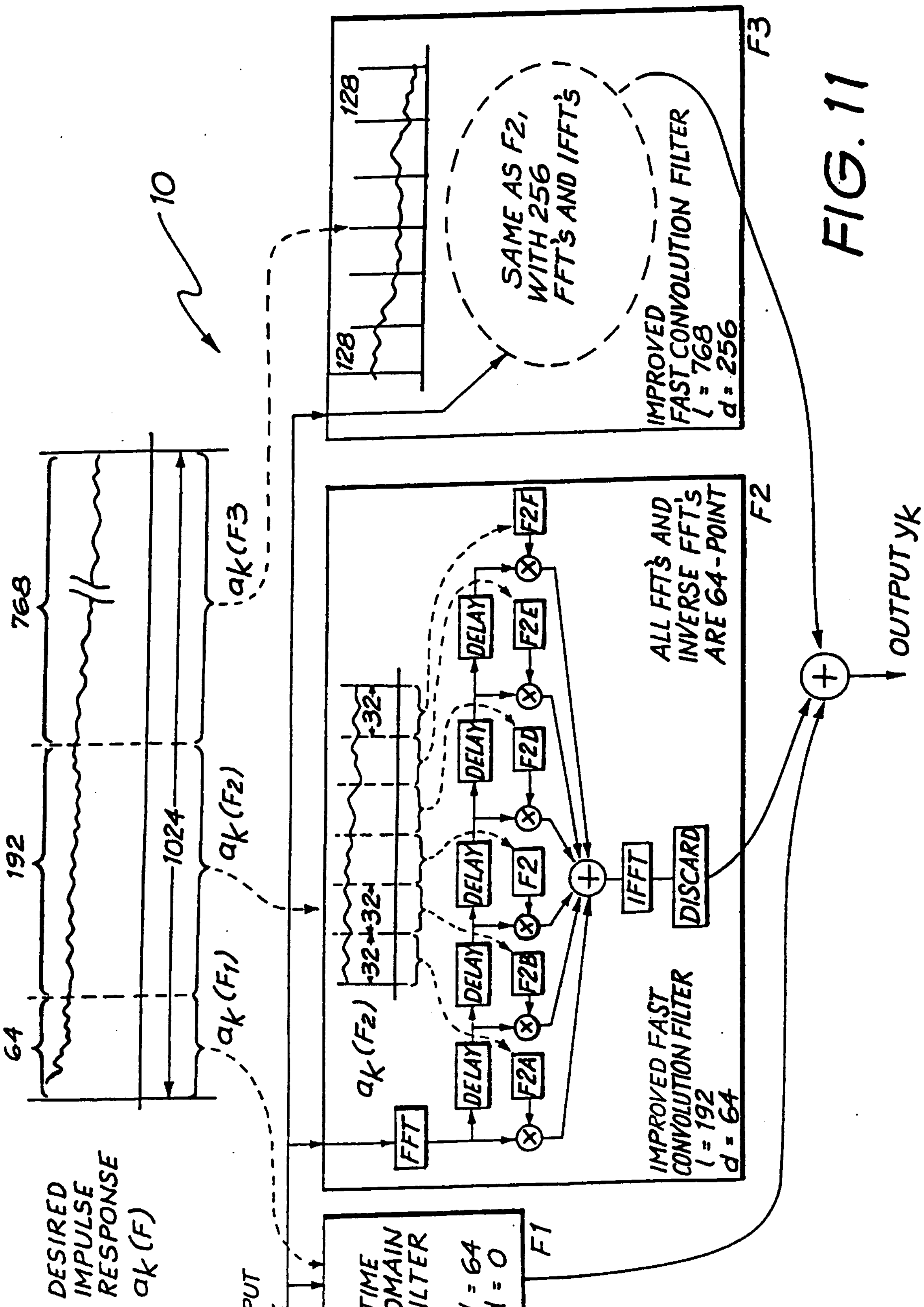
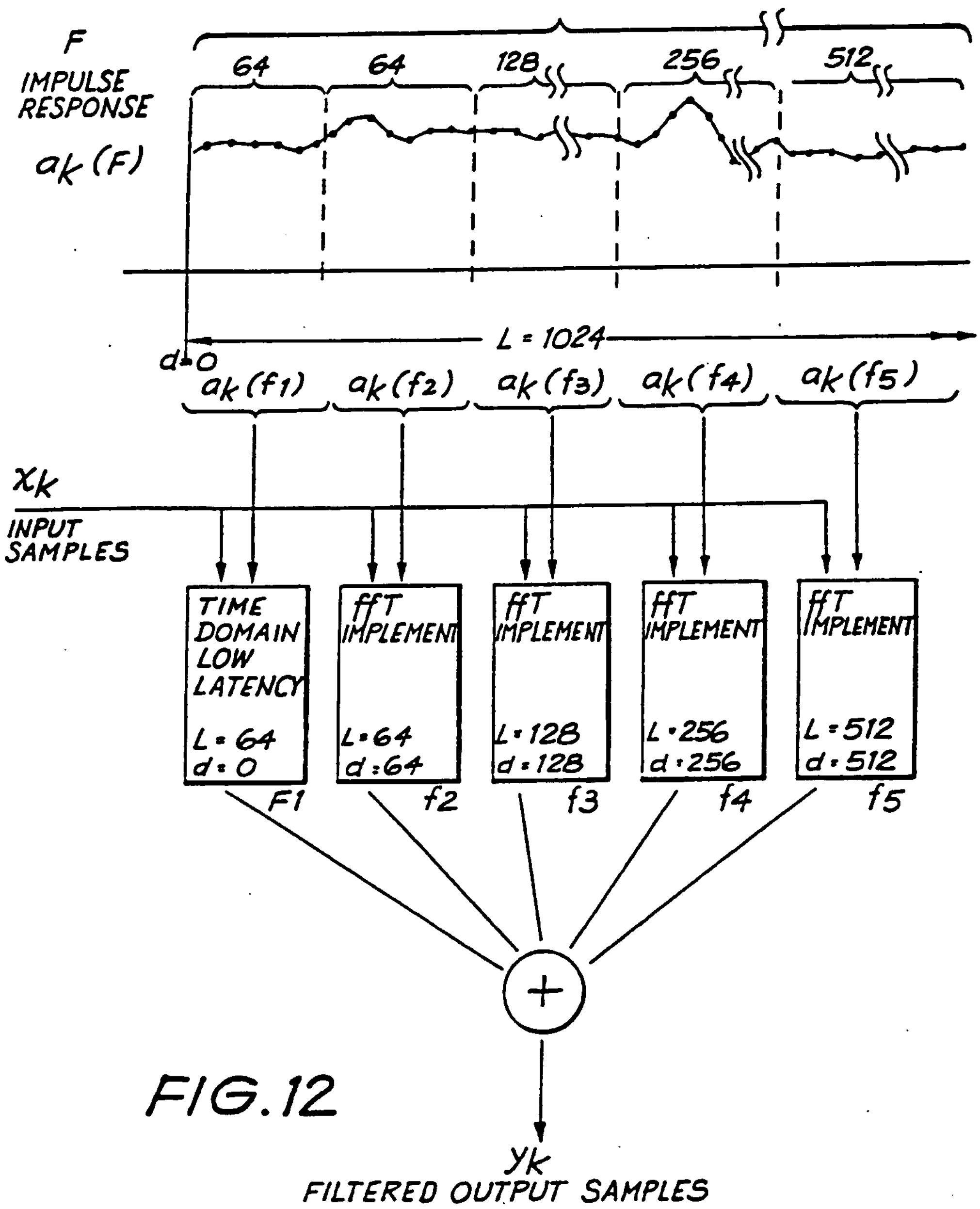
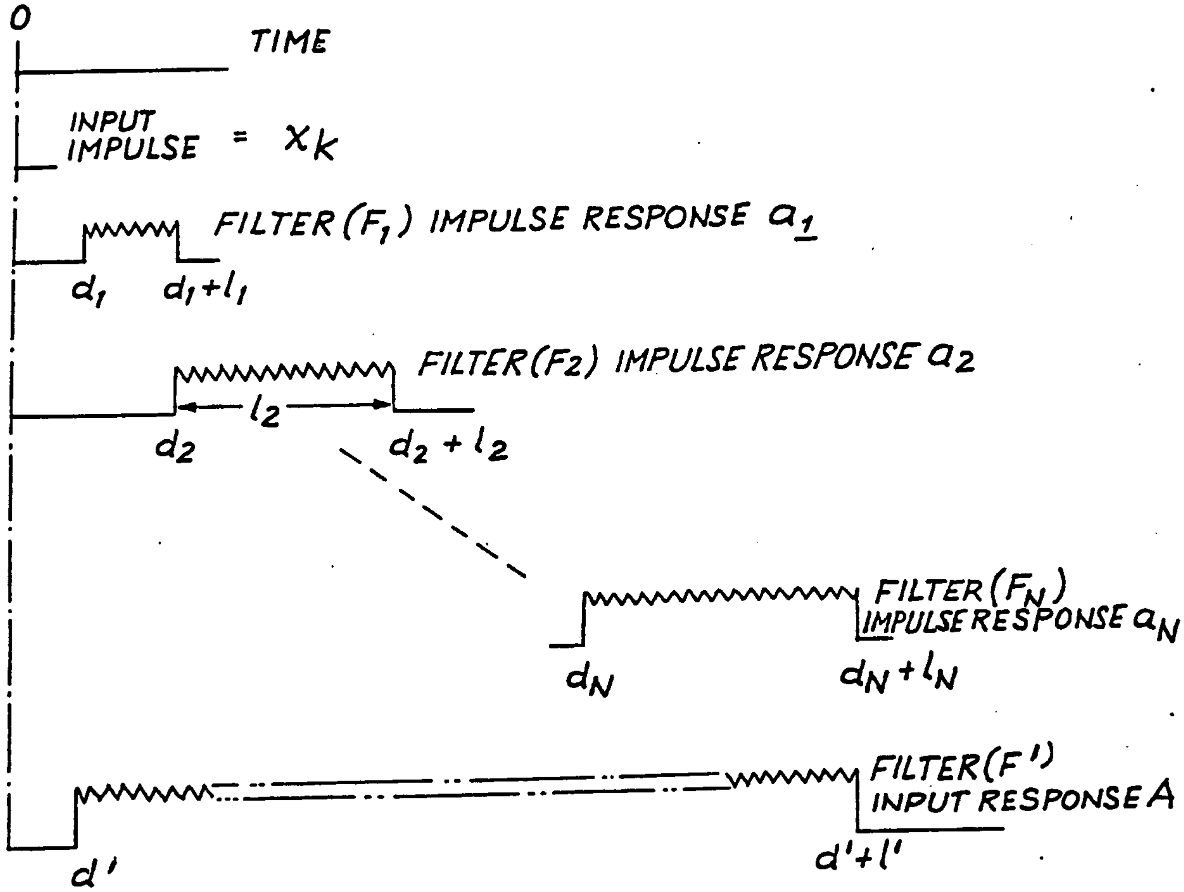

10
DESIRED IMPULSE RESPONSE ak (F)
64
192
768
1024
ak (F1)
ak (F2)
ak (F3)
INPUT xk
TIME DOMAIN FILTER
l = 64
d = 0
F1
FFT
DELAY
F2A
F2B
F2
F2D
F2E
F2F
IFFT
DISCARD
ak (F2)
32
IMPROVED FAST CONVOLUTION FILTER
l = 192
d = 64
ALL FFT'S AND INVERSE FFT'S ARE 64-POINT
F2
128
SAME AS F2, WITH 256 FFT'S AND IFFT'S
IMPROVED FAST CONVOLUTION FILTER
l = 768
d = 256
F3
OUTPUT yk

FIG. 11



F
IMPULSE RESPONSE
$a_k(F)$
64
64
128
256
512
$d=0$
$L=1024$
$a_k(f1)$
$a_k(f2)$
$a_k(f3)$
$a_k(f4)$
$a_k(f5)$
x_k
INPUT SAMPLES
TIME DOMAIN LOW LATENCY
L=64
d=0
F1
ffT IMPLEMENT
L=64
d=64
f2
ffT IMPLEMENT
L=128
d=128
f3
ffT IMPLEMENT
L=256
d=256
f4
ffT IMPLEMENT
L=512
d=512
f5
+
y_k
FILTERED OUTPUT SAMPLES

FIG.12



THE FILTER F' IS MADE BY SUMMING TOGETHER THE OUTPUTS OF FILTERS $F_1, F_2, \ldots\ldots\ldots, F_N$.

EACH FILTER F_i DEFINES OUTPUTS $y_{i,k}$ THE OUTPUT OF FILTER F_i AT TIME SAMPLE k, AS BEING

$$y_{i,k} = \sum_{n=d_i}^{d_i+l_i-1} a_{i,n}\, x_{k-n}$$

IF WE SET $y'_k = \sum_{n=1}^{N} y_{i,k}$

THEN WE HAVE $y'_k = \sum_{n=d'}^{d'+l'-1} A_n\, x_{k-n}$

FIG. 13

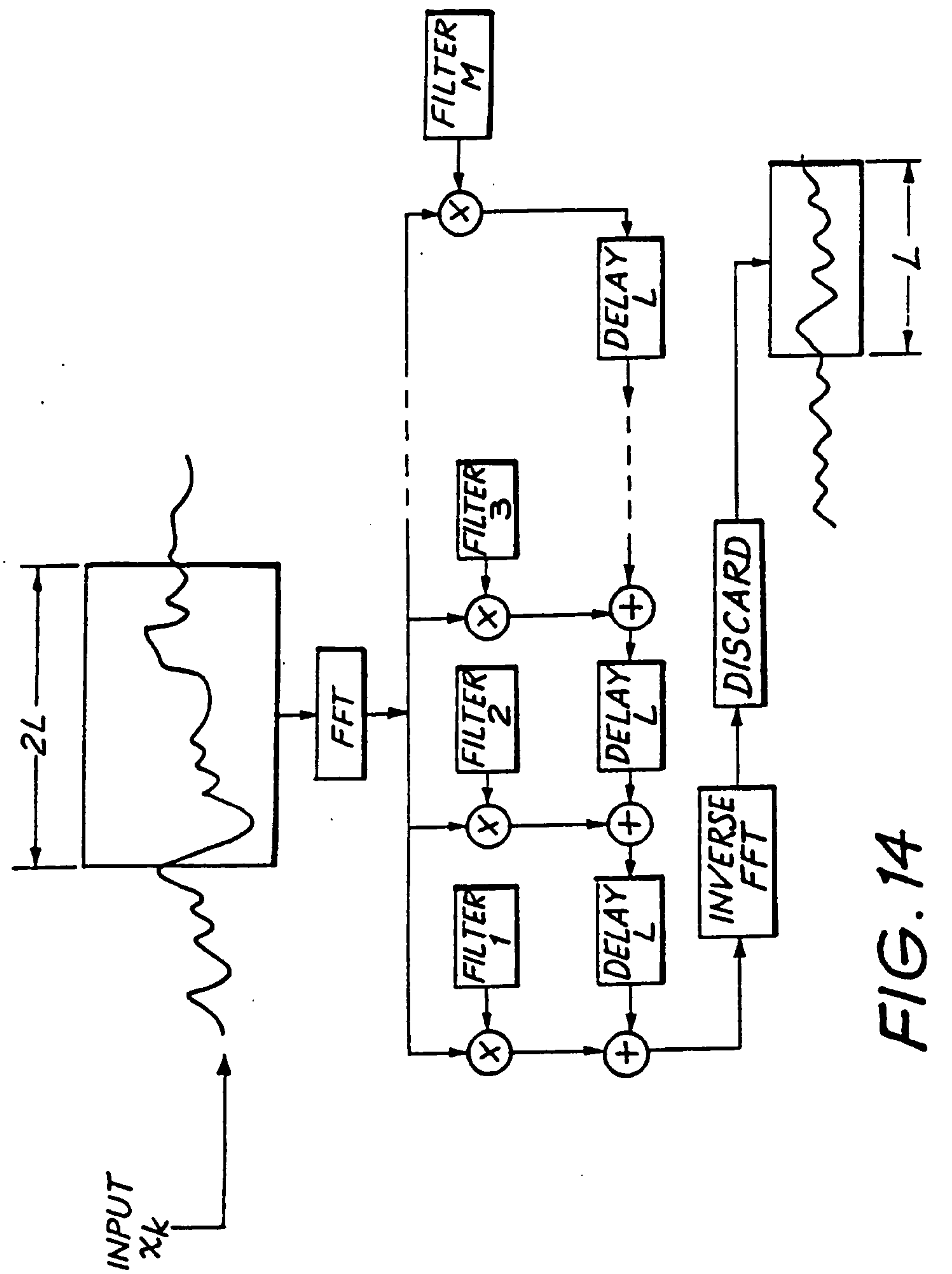

INPUT
x_k
2L
FFT
FILTER 1
FILTER 2
FILTER 3
FILTER M
DELAY L
DELAY L
DELAY L
INVERSE FFT
DISCARD
L

FIG. 14

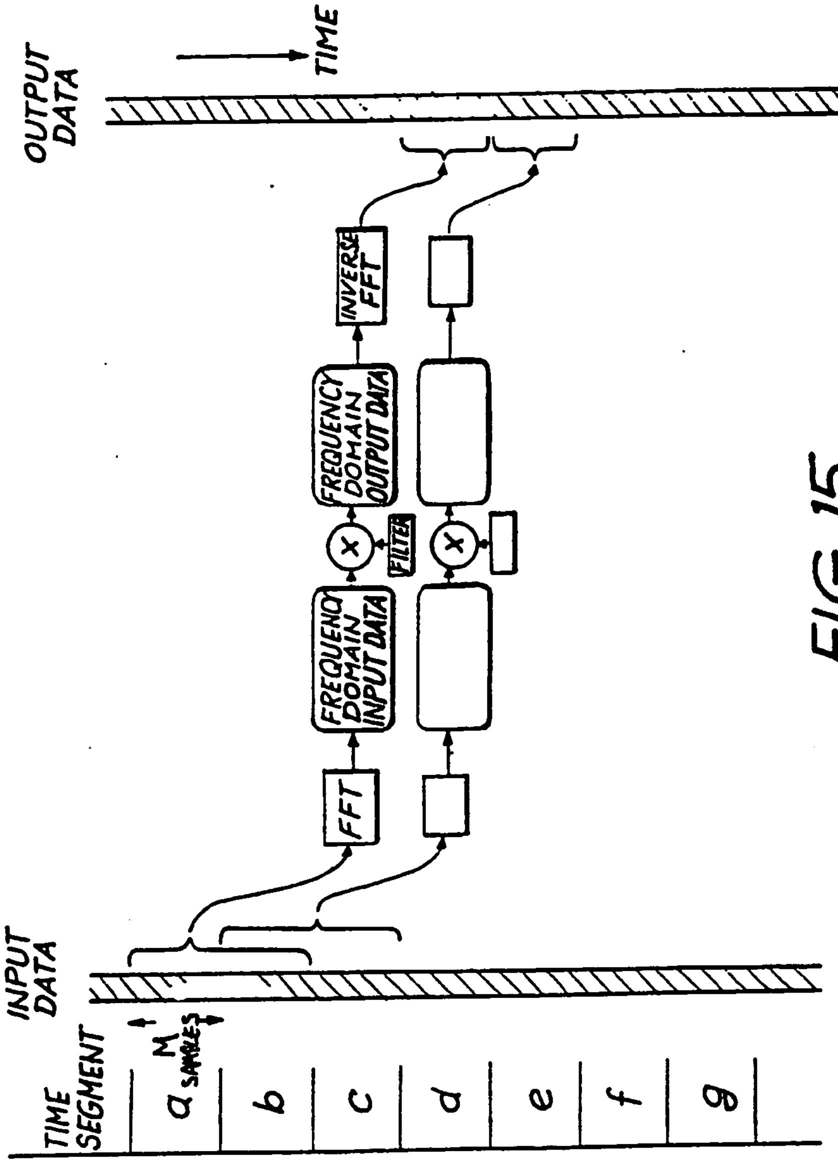

OUTPUT DATA
TIME
INVERSE FFT
FREQUENCY DOMAIN OUTPUT DATA
FILTER
FREQUENCY DOMAIN INPUT DATA
FFT
INPUT DATA
TIME SEGMENT
M SAMPLES
a
b
c
d
e
f
g

FIG. 15

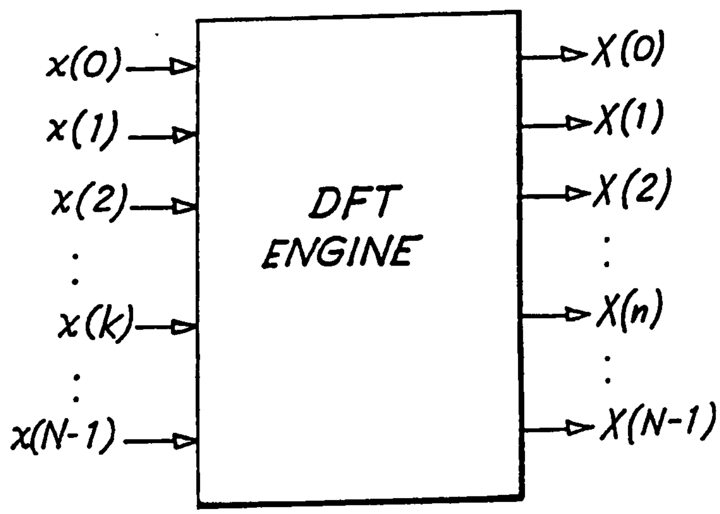


FIG. 16

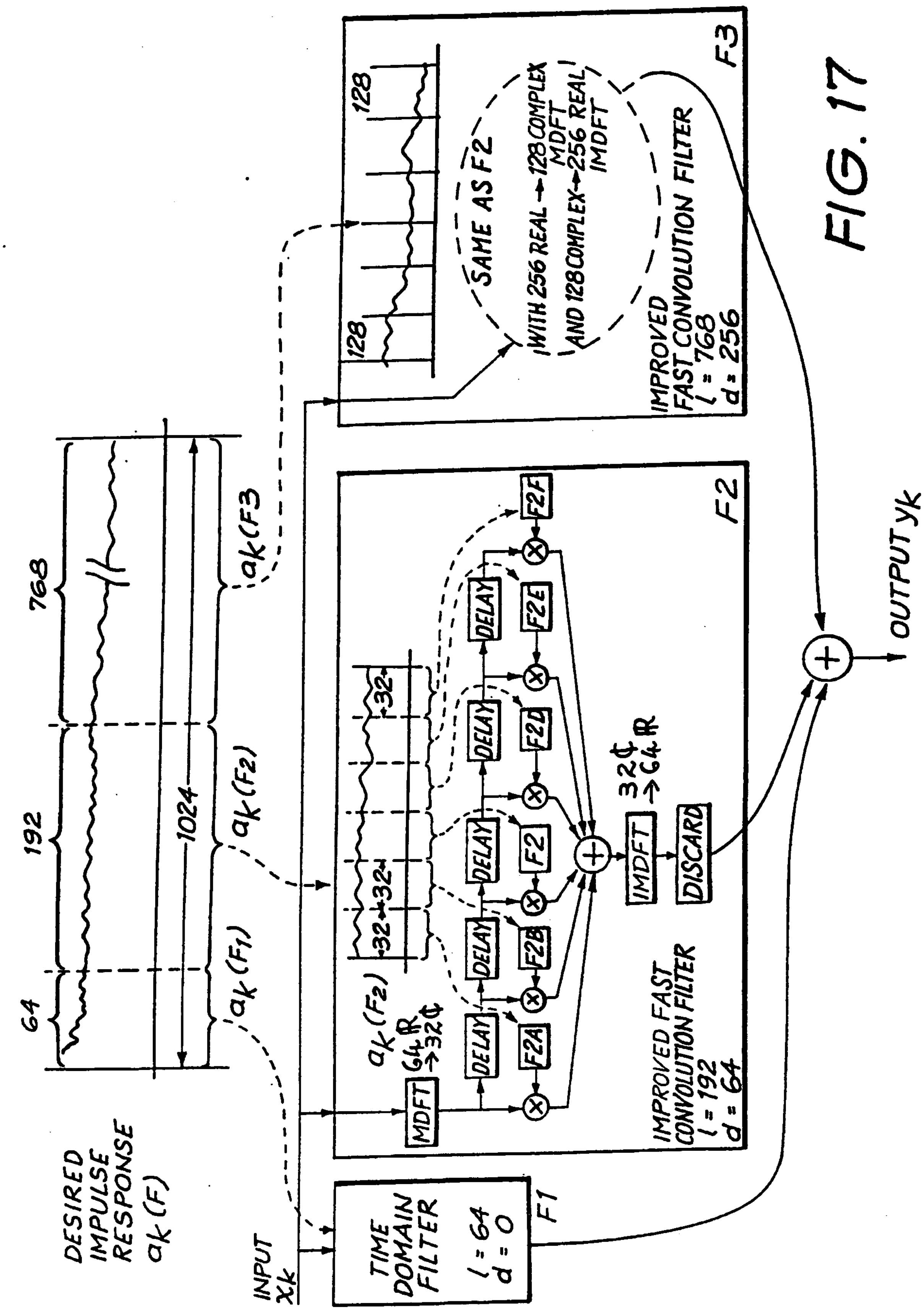

DESIRED IMPULSE RESPONSE $a_k(F)$
64
192
768
1024
$a_k(F_1)$
$a_k(F_2)$
$a_k(F3)$
INPUT x_k
TIME DOMAIN FILTER
$l = 64$
$d = 0$
F1
MDFT
$a_k(F_2)$ 64 R → 32 ¢
DELAY
F2A
F2B
F2
F2D
F2E
F2F
32
32¢ → 64 R
IMDFT
DISCARD
IMPROVED FAST CONVOLUTION FILTER
$l = 192$
$d = 64$
F2
128
SAME AS F2
WITH 256 REAL → 128 COMPLEX MDFT
AND 128 COMPLEX → 256 REAL IMDFT
IMPROVED FAST CONVOLUTION FILTER
$l = 768$
$d = 256$
F3
OUTPUT y_k

FIG. 17

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU 93/00330

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl.^{5} H03H 17/6, 17/2, G06F 15/332

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC: H03H 17/6, 17/2, 7/28, G06F 15/332, 15/33

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
AU: IPC as above

Electronic data base consulted during the international search (name of data base, and where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to Claim No.
X Y	US,A,4992967 (AUVRAY) 12 February 1991 (12.02.91) Claims 1-3, Figs. 9-10 Whole document	19-25,27-28 1-2,6-10,14-18
X	EP,A,250048 (N.V. PHILLIPS' GLOELLAMPENFABRIEKEN GROENEWOUDSEWEG) 23 December 1987 (23.12.87) Page 4 lines 1-20, Fig. 1	19-25,27-28
X	EP,A,448758 (STANDARD ELEKTRIK LORENZ AKTIENGESELLSCHAFT) 2 October 1991 (02.10.91) Claim 1	27,28

[x] Further documents are listed in the continuation of Box C.

[x] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 September 1993 (17.09.93)	30 SEP 1993 (30.09.93)

Name and mailing address of the ISA/AU

AUSTRALIAN INDUSTRIAL PROPERTY ORGANISATION
PO BOX 200
WODEN ACT 2606
AUSTRALIA

Facsimile No. (06) 2853929

Authorized officer

R. CHIA

Telephone No. (06) 2832185

C(Continuation).	DOCUMENTS CONSIDERED TO BE RELEVANT	
Category*	Citation of document, with indication, where appropriate of the relevant passages	Relevant to Claim No.
	WO,A,88/03341 (FUJITSU LIMITED) 5 May 1988 (05.05.88)	
X	Page 11 lines 2-30	19-25,27-28
Y	Page 14 lines 1-10	1-2,6-10,14-16
	US,A,4623980 (VARY) 18 November 1986 (18.11.86)	
X	Figs. 1-2, column 3 lines 10-40	19-25,27-28
Y	Column 2 lines 59-65	17-18
	Proceedings of The IEEE, Vol. 75, No. 9, issued September 1987 (New York), R.C. Agarwal, "Vectorized Mixed Radix Discrete Fourier Transform Algorithms", pages 1283-1292	
X	Whole document	33

Box I Observations where certain claims were found unsearchable (Continuation of Item 1 of first sheet)

This international search report has not established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claim Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. Claims 1-32
Apparatus for and method of performing digital filtering,

2. Claim 33
Fourier transform processor for transforming strings of real numbers, as reasoned on the extra sheet.

(See Box II continued)

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims

2. ☒ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

(continuation) BOX II

The international application does not comply with the requirements of unity of invention because it does not relate to one invention or to a group of inventions so linked as to form a single general inventive concept.

In coming to this conclusion the International Searching Authority has found that there are two inventions:

1. Claims 1-32 directed to digital filtering of sampled data by operating a number of component filters in parallel, each associated with a different delay. It is considered that the combination of the outputs of the parallel filter components by addition comprises a first "special technical feature".

2. Claim 32 directed to a Fourier transform processor for transforming string of real numbers. It is considered that the application of a particular algorithm to create a complex vector from a series of input data comprises a second separate "special technical feature".

Since the abovementioned groups of claims do not share either of the technical features identified, a "technical relationship" between the inventions, as defined in PCT Rule 13.2 does not exist. Accordingly the international application does not relate to one invention or to a single inventive concept.